

Localização de Letreiros Usando Técnicas Estatísticas

BENANTE, Bruno Alves^{1,2}, MORETTO, Thiago Galves^{1,3},
PEREIRA, Mauro Conti^{1,2,3,4}, PISTORI, Hemerson^{1,3}; KASSAB Jr., Fuad⁴

¹Grupo de Pesquisa em Engenharia e Computação – UCDB, Universidade Católica Dom Bosco, Av. Tamandaré 6000, Campo Grande, MS, Brasil, 79117-900, 55-67-3312-3528; ²Curso de Engenharia Mecatrônica – UCDB, ³Curso de Engenharia de Computação – UCDB, ⁴LAC – Laboratório de Automação e Controle, Departamento de Telecomunicações e Controle, Escola Politécnica da USP, Avenida Prof. Luciano Gualberto, travessa3 nº 158 - CEP - 05508-900, São Paulo, SP, Brasil.

Emails: brunobenante@terra.com.br, thiago.moretto@gmail.com, mauro@ucdb.br, pistori@ucdb.br, fuad@lac.usp.br

Resumo

Este trabalho apresenta resultados para ampliar a funcionalidade de um sistema que passa informações visuais a cegos visando facilitar sua orientação e mobilidade. Estas informações são imagens simplificadas de fotografias capturadas em computador, detectando o contorno e reduzindo seu tamanho, que são então desenhadas na pele do usuário por meio de estimulação eletro-táctil. Aproveitando-se da existência da imagem capturada, foi desenvolvido um software para detecção de letreiros nesta imagem, para que seja possível ao usuário perceber letreiros de portas e sinais, desenhando-se o contorno destas letras e números em sua pele.

Palavras chave: Processamento de imagens, técnicas estatísticas, substituição sensorial, localização de letreiros

1. Introdução

Na USP e UCDB foi desenvolvido um sistema que passa informações visuais a cegos visando facilitar sua orientação e mobilidade, capturando uma imagem, processando-a para detectar seu contorno e reduzir sua resolução, e depois desenhar esta imagem simplificada na pele do usuário por meio de eletro-estimulação táctil, ajudando-o a montar um mapa mental de um ambiente desconhecido.

Este trabalho apresenta resultados para ampliar a funcionalidade deste sistema. Como imagem já foi capturada, desenvolveu-se um software para detecção de letreiros nesta imagem, para que seja possível ao usuário perceber letreiros de portas e sinais, desenhando-se o contorno destas letras e números em sua pele.

O algoritmo desenvolvido usa conceitos estatísticos para encontrar caracteres em uma imagem. A imagem é capturada via webcam ligada a um PC, ou a um sensor de imagem CCD ou MOS ligado a um microcontrolador. Na *webcam* a

resolução é de 320 x 240 pixels, cada qual representado por 3 bytes individuais, um para cada cor (RGB, vermelho, verde e azul).

A primeira etapa consiste em transformar a imagem em tons de cinza, e depois binarizá-la. Em vez do tradicional algoritmo de Otsu, global, usa-se algoritmo de binarização regional de Niblack, para compensar eventual variação de luminosidade na imagem.

Para encontrar o letreiro, procura-se por regiões regulares que podem indicar trechos de caracteres, procurando a região onde ocorra maior alternância de faixas pretas e brancas, indicando possível linha contendo letras.

Para isso, primeiro usou-se “*run length encoding*”, representando cada linha da imagem como seqüência de números indicando a quantidade de pontos pretos e brancos alternadamente. Foram então calculados para cada linha, a média e o desvio padrão dos segmentos de pontos pretos, gerando um vetor onde cada posição corresponde a uma linha da imagem. Extrai-se também uma média de todas as médias das linhas e uma média de todos os desvios padrões das linhas. Quatro condições são então verificadas para localizar a região onde ocorre maior alternância de faixas pretas e brancas, a linha com maior chance para conter letras.

Para verificar então onde começa e termina a linha, poderíamos repetir o processo anterior, agora para as colunas. Mas é mais simples eliminar as colunas que contenham apenas pontos brancos, indo da esquerda para direita, até achar o início das letras, e depois da direita para a esquerda, até achar o final do letreiro. O processo poderia ser feito também considerando caracteres brancos em fundo preto, mas os resultados obtidos neste caso foram piores.

Ao trabalhar com a estimulação cutânea e visto que o tato tem uma série de deficiências em relação à visão, um pré-processamento se faz necessário. No caso, o tato tem menor densidade de receptores nervosos e menor representação cortical, ao passo que a visão apresenta na retina uma camada de células que detectam os

contornos dos objetos. O pré-processamento então tenta compensar essas deficiências detectando contornos e reduzindo a resolução da imagem a ser desenhada, assim que capturadas são convertidas de RGB para tons de cinza, tem suas bordas detectadas, é binarizada e sua resolução diminuída, mas estes algoritmos são descritos em outro trabalho Pereira (2006).

Vale destacar que o sistema não contempla, pelo menos ainda, um sistema de reconhecimento óptico de caracteres (OCR) para descobrir quais são as letras e números do letreiro, apenas desenha-se o contorno destes permitindo que o próprio usuário, após o treinamento, consiga reconhecê-los.

2. Metodologia

Visto que o foco do trabalho visa à localização de um letreiro em uma determinada imagem capturada, o processo utilizado para o tratamento é semelhante aos usados em reconhecimento de placas de veículos. Porém, em veículos existem padrões pré-definidos que facilitam a solução do problema, tais como, o tamanho da placa, altura do solo à placa, tamanho e espaçamento entre letras.

Por exemplo, Ketelaars (2002) segmenta a imagem, passando-a para tons de cinza e depois extraindo as bordas, e após tais processos faz uma contagem de pixels pretos em cada linha. Como o fundo da imagem está branco e as bordas em preto, as linhas que contiverem mais bordas, ou seja, mais pixels pretos, são uma das prováveis regiões da placa. Este processo é feito para as colunas com o mesmo intuito. Porém esta solução não se aplica aqui, pois se uma imagem contiver um desenho com uma variação muito grande, irá influenciar no resultado do programa podendo não localizar o letreiro.

Palma (2004) retrata vários tipos de segmentações e alguns métodos para identificação dos letreiros, entre os quais um que faz a segmentação extraindo regiões conexas, definindo-as como componentes conexas, homogêneos em termos da sua luminância extraídos de uma ou mais imagens pré-processadas adequadamente de uma imagem inicial. Entretanto, o seu pré-processamento é feito de forma a que os caracteres surjam como regiões homogêneas. Este processo é dividido em 3 partes, nas quais é feita a normalização da intensidade de luminância, a binarização e a extração dos componentes conexas. Após este processo, classificam-se as regiões utilizando regras de área, altura, largura, proximidade, excentricidade, solidez e contraste por técnicas propostas por ele. As regiões em que se verificam todas estas regras são classificadas como texto.

Neste trabalho optou-se por trabalhar com a imagem binária, mas a rotina de binarização do MATLAB baseia-se no método de limiarização global de Otsu, como visto em Mathworks (2003). Este método não se mostra adequado quando há muita variação de luminosidade na imagem, requerendo uma binarização regional, tendo sido criada então uma implementação do método de Niblack visto em De Souza (2000a).

Foi também implementado um algoritmo que por sua vez é baseado em relações estatísticas. Ele consiste em calcular a média (1) e o desvio padrão (2) entre os pixels vizinhos, e depois comparar o valor analisado com o limiar $T(x,y)$ (3). Se $T(x,y)$ for maior que o *pixel* analisado, é considerado pertencente a algum objeto, caso contrário é considerado pertencente ao fundo como visto em De Souza (2000a),(2000b) e Vasconcelos (2001).

$$(1) \quad \mu(x, y) = \frac{1}{M.N} \sum_{i=0}^M \sum_{j=0}^N p(x, y)$$

$$(2) \quad \sigma(x, y) = \frac{1}{M.N} \sum_{i=0}^M \sum_{j=0}^N (p(x, y) - \mu(x, y))^2$$

$$(3) \quad T(x, y) = -\alpha.\sigma(x, y) + \mu(x, y)$$

A equação (3) nos mostra T , que é o Limiar sendo calculado através de α , que controla a influência do desvio padrão sobre a média da janela, multiplicado pelo desvio padrão somado à média. O índice α foi estimado em 0.2 como visto em De Souza (2000a), sendo o valor que proporcionou melhor resultado para a busca do letreiro na imagem.

Após o procedimento de adquirir e binarizar a imagem, foi preciso definir como detectar o letreiro. Utilizando conceitos estatísticos, como feitos no processo anterior, verificou-se que a melhor forma de localizar um letreiro era procurando regiões regulares que poderiam indicar trechos de caracteres.

Para isso, primeiro usou-se "*run length encoding*", representando cada linha da imagem como uma seqüência de números indicando a quantidade de pontos pretos e brancos alternadamente. Foi então calculada para cada linha a média e o desvio padrão dos segmentos de pontos pretos, gerando um vetor.

As fórmulas utilizadas são semelhantes às do processo anterior (binarização por Niblack). É calculado o número de pixels pretos gerando um vetor onde cada posição corresponde a uma linha da imagem, extraindo também uma média de todas as médias das linhas e uma média de todos os desvios padrões das linhas.

Após os cálculos, são estabelecidas quatro condições para localizar a possível região do letreiro sendo elas listadas abaixo:

Condição 1

$$\left(\frac{Np(x)}{DP(x)} \geq MediaDP \right) \& \& \left(\frac{Np(x)}{DP(x)} > 0 \right)$$

Condição 2

$$\left(\frac{Np(x)}{MD(x)} \geq MediaMD \right) \& \& \left(\frac{Np(x)}{MD(x)} > 0 \right)$$

Condição 3

$$\left(\frac{Coluna}{DP(x)} \geq MediaDP \right) \& \& \left(\frac{Coluna}{DP(x)} > 0 \right)$$

Condição 4

$$\left(\frac{Coluna}{MD(x)} \geq MediaMD \right) \& \& \left(\frac{Coluna}{MD(x)} > 0 \right)$$

Estas quatro condições são estabelecidas no software, na ordem mostrada acima, em que:

- Np é o vetor de quantidade de pixels, cada elemento corresponde a uma linha;
- DP é o vetor de desvio padrão, cada elemento corresponde a uma linha;
- MediaDP é a média dos desvios padrões de cada linha;
- MD é o vetor de média das linhas;
- MediaMD é a média das médias de todas as linhas;
- Coluna é a quantidade de colunas existente na imagem e não um vetor.

O algoritmo entra na primeira condição e checa a saída, que neste caso seria uma imagem. Se não retornar uma resposta, serão verificadas as três condições restantes. Isto estaria forçando uma resposta para a saída, e caso ele não retorne nada de saída na chamada da instrução, é feito um reprocessamento gerando uma nova matriz da imagem que contenha somente as linhas que apresentam o desvio padrão abaixo da média dos desvios padrões. Porém volta-se novamente ao início do processamento com a nova imagem binária extraída da anterior e, refaz-se novamente todo o processo dos cálculos estatísticos.

Depois de localizado o letreiro, é feita uma verificação para identificar onde começam e terminam os caracteres.

Poderia ser aplicado um método semelhante para as colunas, mas como a imagem é binarizada, aplica-se uma heurística mais simples. Seguindo da esquerda para a direita, eliminam-se as colunas iniciais que contenham apenas pontos brancos, indicando que ainda não começaram os caracteres, armazenando esta imagem. Aplica-se então a mesma idéia da direita para a esquerda, eliminando as colunas em branco até encontrar a última que contenha pontos brancos, armazenando a imagem resultante.

Em alguns casos, as condições também podem ser verificadas com a contagem dos pixels brancos. Mas como o fundo da imagem é branco e

as bordas pretas procuram-se então a seqüência de pixels pretos, segundo a idéia de Ketelaars (2002) de que “um caractere é relativamente um ponto preto em um fundo branco”. A contagem com preto é mais viável e mostrou uma abrangência maior de situações para localização do letreiro.

Estes cálculos estatísticos foram escolhidos porque uma maneira fácil de detectar a variação de pixels é através do desvio padrão. Estas variações ocorrem nas regiões onde se encontram os letreiros, possuindo comprimento de segmentos de uns e zeros de tamanhos quase iguais. Isto faz com que a média entre eles diminua e por consequência o desvio padrão também.

3. Resultados e Discussão

Para o exemplo da Figura 1, cujo resultado da binarização por Niblack é mostrado na Figura 2, é calculado o desvio padrão por linha, mostrado na Figura 3. Na região do letreiro, o gráfico apresenta em quase todas as situações os valores dos desvios padrões baixos, facilitando identificar estas linhas, destacadas com um círculo vermelho.

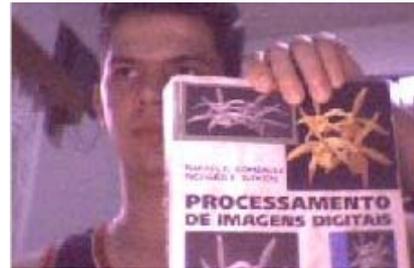


Fig.1 Imagem Original



Fig.2 Imagem Binarizada

Estas posições no gráfico estão aproximadamente entre as linhas 195 a 220, sendo as possíveis linhas que uma das quatro condições detecta. Feita a análise das colunas do lado esquerdo e direito, é gerada a imagem da Figura 4.

Quanto aos casos que necessitam de reprocessamento, pode-se observar um exemplo no qual o software refaz a imagem a partir da imagem de tamanho original binarizada, que contém as linhas com o desvio padrão abaixo da média dos desvios padrões. Segue um exemplo no qual foi feito um reprocessamento.

A imagem da Figura 5 foi binarizada gerando a Figura 6, e o algoritmo não conseguiu gerar uma saída, então o sistema automaticamente reprocessa tudo, construindo uma nova imagem que é mostrada na Figura 7.

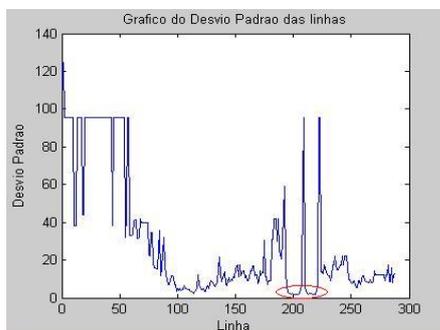


Fig.3 Gráfico do Desvio Padrão do tamanho dos segmentos de cada linha

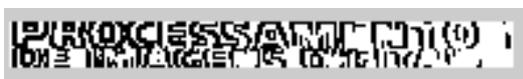


Fig.4 Imagem Final



Fig.5 Imagem Original



Fig.6 Imagem Binarizada



Fig.7 Imagem Final

4. Conclusão

Tanto nos exemplos mostrados, como em outros testes realizados, os letreiros foram localizados através da variação dos pixels, realizando os cálculos estatísticos linha a linha dos desvios padrões do tamanho dos segmentos de zeros e uns e analisando as colunas.

Como este processamento faz parte de um sistema maior com intenção de passar

informações ao cego através de estimulação tátil, pretende-se ainda implementar técnicas de aproximação (*zoom in e zoom out*) para melhorar a visualização ao usuário.

Falta também definir melhor os contornos das letras e separá-las. Para isso existe a idéia de usar morfologia matemática, Salem (2005), para engrossar as linhas dos caracteres, e usar técnicas de *chain code*, Gonzalez (2000), para a geração da seqüência de estimulação de cada eletrodo da matriz aplicada sobre a pele.

5. Referências

De Souza, F.P.C. (2000) "Localização e Leitura Automática de Caracteres Alfanuméricos - Uma Aplicação na Identificação de Veículos". Dissertação de Pós-Graduação", UFRGS, Porto Alegre, RS, Brasil.

De Souza, F.P.C, Suzin, A. (2000) *SIAV-Um Sistema de Identificação automática de Veículos*. XIII Congresso Brasileiro de Automática, Florianópolis - SC, UFRGS, p 1377-1380.

Gonzalez, R. C. e Woods, R.E. (2000) *Processamento de Imagens Digitais*. Editora Edgard Blucher Ltda, São Paulo, Brasil.

Ketelaars, N.(2002) *Automated License Plate Recognition*. *AIME Magazine*, p 09-12.

Mathworks Inc. (2003) *MatLab's Image Processing Toolbox Manual*. Mathworks. Cambridge, MA.

Palma, D.M.C. (2004) *Extração Automática de Texto em Sequências de Vídeo*. Dissertação de Mestrado, Universidade Técnica de Lisboa. Lisboa, Portugal.

Pereira, M.C. (2006) *Sistema de Substituição Sensorial para Auxílio a Deficientes Visuais Via Técnicas de Processamento de Imagens e Estimulação Cutânea*. Tese de Doutorado, USP, São Paulo, SP, Brasil.

Salem, A.-B. M., Sewisy, A. A., and Elyan, U. A. (2005) *A vertex chain code approach for image recognition*. *ICGST International Journal on Graphics, Vision and Image Processing*.

Vasconcelos, C.A.D., Pereira, T.C. (2001) *Implementação de um Módulo localizador de Placas em Imagens Digitalizadas de Veículos*. Projeto de Graduação, UCDB, Campo Grande, MS, Brasil.